

Solving simple stochastic games with few coin toss positions*

Rasmus Ibsen-Jensen
Department of Computer Science
Aarhus University

Peter Bro Miltersen
Department of Computer Science
Aarhus University

March 21, 2012

Abstract

Gimbert and Horn gave an algorithm for solving simple stochastic games with running time $O(r!n)$ where n is the number of positions of the simple stochastic game and r is the number of its coin toss positions. Chatterjee *et al.* pointed out that a variant of strategy iteration can be implemented to solve this problem in time $4^r r^{O(1)} n^{O(1)}$. In this paper, we show that an algorithm combining value iteration with retrograde analysis achieves a time bound of $O(r2^r(r \log r + n))$, thus improving both time bounds. While the algorithm is simple, the analysis leading to this time bound is involved, using techniques of extremal combinatorics to identify worst case instances for the algorithm.

1 Introduction

Simple stochastic games is a class of two-player zero-sum games played on graphs that was introduced to the algorithms and complexity community by Condon [6]. A simple stochastic game is given by a directed finite (multi-)graph $G = (V, E)$, with the set of vertices V also called *positions* and the set of arcs E also called *actions*. There is a partition of the positions into V_1 (positions belonging to player Max), V_2 (positions belonging to player Min), V_R (coin toss positions), and a special terminal position GOAL. Positions of V_1, V_2, V_R have exactly two outgoing arcs, while the terminal position GOAL has none. We shall use r to denote $|V_R|$ (the number of coin toss positions) and n to denote $|V| - 1$ (the number of non-terminal positions) throughout the paper. Between moves, a pebble is resting at one of the positions k . If k belongs to a player, this player should strategically pick an outgoing arc from k and move the pebble along this arc to another node. If k is a position in V_R , Nature picks an outgoing arc from k uniformly at random and moves the pebble along this arc. The objective of the game for player Max is to reach GOAL and should play so as to maximize his probability of doing so. The objective for player Min is to minimize player Max's probability of reaching GOAL.

A *strategy* for a simple stochastic game is a (possibly randomized) procedure for selecting which arc or action to take, given the history of the play so far. A *positional strategy* is the very special case of this where the choice is deterministic and only depends on the current position, i.e., a

*The authors acknowledge support from the Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, within which this work was performed. The authors also acknowledge support from the Center for Research in Foundations of Electronic Markets (CFEM), supported by the Danish Strategic Research Council.

positional strategy is simply a map from positions to actions. If player Max plays using strategy x and player Min plays using strategy y , and the play starts in position k , a random play $p(x, y, k)$ of the game is induced. We let $u(x, y)$ denote the probability that player Max will reach GOAL in this random play. A strategy x^* for player Max is said to be *optimal* if for all positions k it holds that

$$\inf_{y \in S_2} u^k(x^*, y) \geq \sup_{x \in S_1} \inf_{y \in S_2} u^k(x, y), \quad (1)$$

where S_1 (S_2) is the set of strategies for player Max (Min). Similarly, a strategy y^* for player Min is said to be optimal if

$$\sup_{x \in S_1} u^k(x, y^*) \leq \inf_{y \in S_2} \sup_{x \in S_1} u^k(x, y). \quad (2)$$

A general theorem of Liggett and Lippman ([12], fixing a bug of a proof of Gillette [9]) restricted to simple stochastic games, implies that:

- Optimal positional strategies x^*, y^* for both players exist.
- For such optimal x^*, y^* and for all positions k ,

$$\min_{y \in S_2} u^k(x^*, y) = \max_{x \in S_1} u^k(x, y^*).$$

This number is called the *value* of position k . We shall denote it $\text{val}(G)_k$ and the vectors of values $\text{val}(G)$.

In this paper, we consider *quantitatively solving* simple stochastic games, by which we mean computing the values of all positions of the game, given an explicit representation of G . Once a simple stochastic game has been quantitatively solved, optimal strategies for both players can be found in linear time [2]. However, it was pointed out by Anne Condon twenty years ago that no worst case polynomial time algorithm for quantitatively solving simple stochastic games is known. By now, finding such an algorithm is a celebrated open problem. Gimbert and Horn [10] pointed out that the problem of solving simple stochastic games parametrized by $r = |V_R|$ is *fixed parameter tractable*. That is, simple stochastic games with “few” coin toss positions can be solved efficiently. The algorithm of Gimbert and Horn runs in time $r!n^{O(1)}$. The next natural step in this direction is to try to find an algorithm with a better dependence on the parameter r . Thus, Dai and Ge [8] gave a *randomized* algorithm with *expected* running time $\sqrt{r!}n^{O(1)}$. Chatterjee *et al.* [4] pointed out that a variant of the standard algorithm of *strategy iteration* devised earlier by the same authors [5] can be applied to find a solution in time $4^r r^{O(1)} n^{O(1)}$ (they only state a time bound of $2^{O(r)} n^{O(1)}$, but a slightly more careful analysis yields the stated bound). The dependence on n in this bound is at least quadratic. The main result of this paper is an algorithm running in time $O(r2^r(r \log r + n))$, thus improving all of the above bounds. More precisely, we show:

Theorem 1 *Assuming unit cost arithmetic on numbers of bit length up to $\Theta(r)$, simple stochastic games with n positions out of which r are coin toss positions, can be quantitatively solved in time $O(r2^r(r \log r + n))$.*

The algorithm is based on combining a variant of *value iteration* [15, 7] with *retrograde analysis* [3, 1]. We should emphasize that the time bound of Theorem 1 is valid *only* for simple stochastic games as originally defined by Condon. The algorithm of Gimbert and Horn (and also the algorithm

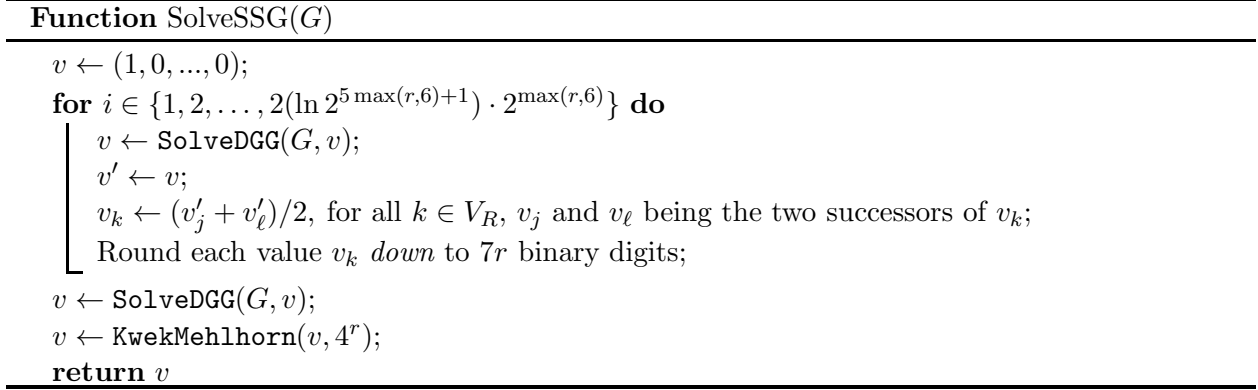


Figure 1: Algorithm for solving simple stochastic games

of Dai and Ge, though this is not stated in their paper) actually applies to a generalized version of simple stochastic games where coin toss positions are replaced with *chance positions* that are allowed arbitrary out-degree and where a not-necessarily-uniform distribution is associated to the outgoing arcs. The complexity of their algorithm for this more general case is $O(r!(|E| + p))$, where p is the maximum bit-length of a transition probability (they only claim $O(r!(n|E| + p))$, but by using retrograde analysis in their Proposition 1, the time is reduced by a factor of n). The algorithm of Dai and Ge has analogous expected complexity, with the $r!$ factor replaced with $\sqrt{r}!$. While our algorithm and the strategy improvement algorithm of Chatterjee *et al.* can be generalized to also work for these generalized simple stochastic games, the dependence on the parameter p would be much worse - in fact exponential in p . It is an interesting open problem to get an algorithm with a complexity polynomial in 2^r as well as p , thereby combining the desirable features of the algorithms based on strategy iteration and value iteration with the features of the algorithm of Gimbert and Horn.

1.1 Organization of paper

In Section 2 we present the algorithm and show how the key to its analysis is to give upper bounds on the difference between the value of a given simple stochastic game and the value of a *time bounded version* of the same game. In Section 3, we then prove such upper bounds. In fact, we offer two such upper bounds: One bound with a relatively direct proof, leading to a variant of our algorithm with time complexity $O(r^2 2^r (r + n \log n))$ and an *optimal* bound on the difference in value, shown using techniques from extremal combinatorics, leading to an algorithm with time complexity $O(r 2^r (r + n \log n))$. In the Conclusion section, we briefly sketch how our technique also yields an improved upper bound on the time complexity of the strategy iteration algorithm of Chatterjee *et al.*

2 The algorithm

2.1 Description of the algorithm

Our algorithm for solving simple stochastic games with few coin toss positions is the algorithm of Figure 1.

Procedure ModifiedValueIteration(G)

```
 $v \leftarrow (1, 0, \dots, 0);$   
while true do  
   $v \leftarrow \text{SolveDGG}(G, v);$   
   $v' \leftarrow v;$   
   $v_k \leftarrow (v'_j + v'_\ell)/2$ , for all  $k \in V_R$ ,  $v_j$  and  $v_\ell$  being the two successors of  $v_k$ ;
```

Figure 2: Modified value iteration

In this algorithm, the vectors v and v' are real-valued vectors indexed by the positions of G . We assume the GOAL position has the index 0, so $v = (1, 0, \dots, 0)$ is the vector that assigns 1 to the GOAL position and 0 to all other positions. **SolveDGG** is the retrograde analysis based algorithm from Proposition 1 in Andersson *et al.* [1] for solving *deterministic graphical games*. Deterministic graphical games are defined in a similar way as simple stochastic games, but they do not have coin toss positions, and arbitrary real payoffs are allowed at terminals. The notation **SolveDGG**(G, v') means solving the deterministic graphical game obtained by replacing each coin toss position k of G with a terminal with payoff v'_k , and returning the value vector of this deterministic graphical game. Finally, **KwekMehlhorn** is the algorithm of Kwek and Mehlhorn [11]. **KwekMehlhorn**(v, q) returns a vector where each entry v_i in the vector v is replaced with the smallest fraction a/b with $a/b \geq v_i$ and $b \leq q$.

The complexity analysis of the algorithm is straightforward, given the analyses of the procedures **SolveDGG** and **KwekMehlhorn** from [1, 11]. There are $O(r2^r)$ iterations, each requiring time $O(r \log r + n)$ for solving the deterministic graphical game. Finally, the Kwek-Mehlhorn algorithm requires time $O(r)$ for each replacement, and there are only r replacements to be made, as there are only r different entries different from 1, 0 in the vector v , corresponding to the r coin toss positions, by standard properties of deterministic graphical games [1].

2.2 Proof of correctness of the algorithm

We analyse our main algorithm by first analysing properties of a simpler non-terminating algorithm, depicted in Figure 2. We shall refer to this algorithm as *modified value iteration*.

Let v^t be the content of the vector v immediately after executing **SolveDGG** in the $(t + 1)$ 'st iteration of the loop of **ModifiedValueIteration** on input G . To understand this variant of value iteration, we may observe that the v^t vectors can be given a “semantics” in terms of the value of a time bounded game.

Definition 2 Consider the “timed modification” G^t of the game G defined as follows. The game is played as G , except that play stops and player Max loses when the play has encountered $t + 1$ (not necessarily distinct) coin toss positions. We let $\text{val}(G^t)_k$ be the value of G^t when play starts in position k .

Lemma 3 $\forall k, t : v_k^t = \text{val}(G^t)_k$.

Proof Straightforward induction in t (“backwards induction”). \square

From the semantics offered by Lemma 3 we immediately have $\forall k, t : \text{val}(G^t)_k \leq \text{val}(G^{t+1})_k$. Furthermore, it is true that $\lim_{t \rightarrow \infty} \text{val}(G^t) = \text{val}(G)$, where $\text{val}(G)$ is the value vector of G . This

latter statement is very intuitive, given Lemma 3, but might not be completely obvious. It may be established rigorously as follows:

Definition 4 For a given game G , let the game \bar{G}^t be the following. The game is played as G , except that play stops and player Max loses when the play has encountered $t + 1$ (not necessarily distinct) positions. We let $\text{val}(\bar{G}^t)_k$ be the value of \bar{G}^t when play starts in position k .

(We note that $\text{val}(\bar{G}^t)$ is the valuation computed after t iterations of *unmodified* value iteration [7].) A very general theorem of Mertens and Neyman [13] linking the value of an infinite game to the values of its time limited versions implies that $\lim_{t \rightarrow \infty} \text{val}(\bar{G}^t) = \text{val}(G)$. Also, we immediately see that for any k , $\text{val}(\bar{G}^t)_k \leq \text{val}(G^t)_k \leq \text{val}(G)_k$, so we also have $\lim_{t \rightarrow \infty} \text{val}(G^t)_k = \text{val}(G)_k$.

To relate `SolveDGG` of Figure 1 to modified value iteration of Figure 2, it turns out that we want to upper bound the smallest t for which

$$\forall i : \text{val}(G)_i - \text{val}(G^t)_i \leq 2^{-5r}.$$

Let $T(G)$ be that t . We will bound $T(G)$ using two different approaches. The first, in Subsection 3.1, is rather direct and is included to show what may be obtained using completely elementary means. It shows that $T(G) \leq 5(\ln 2) \cdot r^2 \cdot 2^r$, for any game G with r coin toss positions (Lemma 7).

The second, in Subsection 3.2, identifies an *extremal* game (with respect to convergence rate) with a given number of positions and coin toss positions. More precisely:

Definition 5 Let $S_{n,r}$ be the set of simple stochastic games with n positions out of which r are coin toss positions. Let $G \in S_{n,r}$ be given. We say that G is t -extremal if

$$\max_i (\text{val}(G)_i - \text{val}(G^t)_i) = \max_{H \in S_{n,r}} \max_i (\text{val}(H)_i - \text{val}(H^t)_i).$$

We say that G is extremal if it is t -extremal for all t .

It is clear that t -extremal games exists for any choice of n, r and t . (That extremal games exists for any choice of n and r is shown later in the present paper.) To find an extremal game, we use techniques from extremal combinatorics. By inspection of this game, we then get a better upper bound on the convergence rate than that offered by the first approach. We show using this approach that $T(G) \leq 2(\ln 2^{5r+1}) \cdot 2^r$, for any game $G \in S_{n,r}$ (Corollary 14).

Assuming that an upper bound on $T(G)$ is available, we are now ready to finish the proof of correctness of the main algorithm. We will only do so explicitly for the bound on $T(G)$ obtained by the second approach from Subsection 3.2 (the weaker bound implies correctness of a version of the algorithm performing more iterations of its main loop). From Corollary 14, we have that for any game $G \in S_{n,r}$, $\text{val}(G^T)_k$ and hence, modified value iteration, approximates $\text{val}(G)_k$ within an additive error of 2^{-5r} for $t \geq 2(\ln 2^{5r+1}) \cdot 2^r$ and k being any position. `SolveSSG` differs from `ModifiedValueIteration` by rounding down the values in the vector v in each iteration. Let \tilde{v}^t be the content of the vector v immediately after executing `SolveDGG` in the t 'th iteration of the loop of `SolveSSG`. We want to compare $\text{val}(G^t)_k$ with \tilde{v}_k^t for any k . As each number is rounded down by less than 2^{-7r} in each iteration of the loop and recalling Lemma 3, we see by induction that

$$\text{val}(G^t)_k - t2^{-7r} \leq \tilde{v}_k^t \leq \text{val}(G^t)_k.$$

In particular, when $t = 2(\ln 2^{5r+1}) \cdot 2^r$, we have that \tilde{v}_k^t approximates $\text{val}(G)_k$ within $2^{-5r} + 2(\ln 2^{5r+1}) \cdot 2^r 2^{-7r} < 2^{-4r}$, for any k , as we can assume $r \geq 6$ by the code of `SolveSSG` of Figure 1.

Lemma 2 of Condon [6] states that the value of a position in a simple stochastic game with n non-terminal positions can be written as a fraction with integral numerator and denominator at most 4^n . As pointed out by Chatterjee *et al.* [4], it is straightforward to see that her proof in fact gives an upper bound of 4^r , where r is the number of coin toss positions. It is well-known that two distinct fractions with denominator at most $m \geq 2$ differ by at least $\frac{1}{m(m-1)}$. Therefore, since \tilde{v}_k^t approximates $\text{val}(G)_k$ within $2^{-4r} < \frac{1}{4^r \cdot (4^r - 1)}$ from below, we in fact have that $\text{val}(G)_k$ is the smallest rational number p/q so that $q \leq 4^r$ and $p/q \geq \tilde{v}_k^t$. Therefore, the Kwek-Mehlhorn algorithm applied to \tilde{v}_k^t correctly computes $\text{val}(G)_k$, and we are done.

We can not use the bound on $T(G)$ obtained by the first direct approach (in Subsection 3.1) to show the correctness of `SolveSSG`, but we can show the correctness of the version of it that runs the main loop an additional factor of $O(r)$ times, that is, i should range over $\{1, 2, \dots, 5(\ln 2) \cdot r^2 2^r\}$ instead of over $\{1, 2, \dots, 2(\ln 2^{5r+1}) \cdot 2^r\}$.

3 Bounds on the convergence rate

3.1 A direct approach

Lemma 6 *Let $G \in S_{n,r}$ be given. For all positions k and all integers $i \geq 1$, we have*

$$\text{val}(G)_k - \text{val}(G^{i \cdot r})_k \leq (1 - 2^{-r})^i.$$

Proof If $\text{val}(G)_k = 0$, we also have $\text{val}(G^{i \cdot r})_k = 0$, so the inequality holds. Therefore, we can assume that $\text{val}(G)_k > 0$.

Fix some optimal positional strategy, x , for Max in G . Let y be any pure (i.e., deterministic, but not necessarily positional) strategy for Min with the property that y guarantees that the pebble will not reach GOAL after having been in a position of value 0 (in particular, any best reply to any strategy of Max, including x , clearly has this property).

The two strategies x and y together induce a probability space σ_k on the set of plays of the game, starting in position k . Let the probability measure on plays of G^t associated with this strategy be denoted Pr_{σ_k} . Let W_k be the event that this random play reaches GOAL. We shall also consider the event W_k to be a set of plays. Note that any position occurring in any play in W_k has non-zero value, by definition of y .

Claim. There is a play in W_k where each position occurs at most once.

Proof of Claim. Assume to the contrary that for all plays in W_k , some position occurs at least twice. Let y' be the modification of y where the second time a position, v , in V_2 is entered in a given play, y takes the same action as was used the first time v occurred. Let W' be the set of plays generated by x and y for which the pebble reaches GOAL. We claim that W' is in fact the empty set. Indeed, if W' contains any play q , we can obtain a play in W' where each position occurs only once, by removing all transitions in q occurring between repetitions of the same position. Such a play is also an element of W_k , contradicting the assumption that all plays in W_k has a position occurring twice. The emptiness of W' shows that the strategy x does not guarantee that GOAL is reached with positive probability, when play starts in k . This contradicts either that x is optimal or that $\text{val}(G)_k > 0$. We therefore conclude that our assumption is incorrect, and that there is

a play q in W_k where each position occurs only once, as desired. *This completes the proof of the claim.*

The probability according to the probability measure σ_k that a given play where each coin toss position occurs only once occurs, is at least 2^{-r} .

Let W_k^i be the set of plays in W_k that contains at most i occurrences of coin toss positions (and also let W_k^i denote the corresponding event with respect to the measure σ_k). Since the above claim holds for any position k of non-zero value and plays in W_k only visits positions of non-zero value, we see that $\Pr_{\sigma_k}[\neg W_k^{i,r} | W_k] \leq (1 - 2^{-r})^i$, for any i . Since x is optimal, we also have $\Pr_{\sigma_k}[W_k] \geq \text{val}(G)_k$. Therefore,

$$\begin{aligned} \Pr_{\sigma_k}[W_k^{i,r}] &= \Pr_{\sigma_k}[W_k] - \Pr_{\sigma_k}[\neg W_k^{i,r} | W_k] \Pr_{\sigma_k}[W_k] \\ &\geq \text{val}(G)_k - (1 - 2^{-r})^i \end{aligned}$$

The above derivation is true for any y guaranteeing that no play can enter a position of value 0 and then reach GOAL, and therefore it is also true for y being the optimal strategy in the time-limited game, $G^{i,r}$. In that case, we have $\Pr_{\sigma_k}[W_k^{i,r}] \leq \text{val}(G^{i,r})_k$. We can therefore conclude that $\text{val}(G^{i,r})_k \geq \text{val}(G)_k - (1 - 2^{-r})^i$, as desired. \square

Lemma 7 *Let $G \in S_{n,r}$ be given.*

$$T(G) \leq 5(\ln 2) \cdot r^2 \cdot 2^r$$

Proof We will show that for any $t \geq 5(\ln 2) \cdot r^2 \cdot 2^r$ and any k , we have that $\text{val}(G)_k - \text{val}(G^t)_k < 2^{-5r}$.

From Lemma 6 we have that $\forall i, k : \text{val}(G)_k - \text{val}(G^{i,r})_k < (1 - 2^{-r})^i$. Thus,

$$\text{val}(G)_k - \text{val}(G^t)_k < (1 - 2^{-r})^{t/r} = ((1 - 2^{-r})^{2^r})^{\frac{t}{r \cdot 2^r}} < e^{-\frac{t}{r \cdot 2^r}} \leq e^{-\frac{5(\ln 2)r^2 2^r}{r \cdot 2^r}} = 2^{-5r}.$$

\square

3.2 An extremal combinatorics approach

The game of Figure 3 is a game in $S_{n,r}$. We will refer to this game as $E_{n,r}$. $E_{n,r}$ consists of no Max-positions. Each Min-position in $E_{n,r}$ has GOAL as a successor twice. The i 'th coin toss position, for $i \geq 2$, has the $(i - 1)$ 'st and the r 'th coin toss position as successors. The first coin toss position has GOAL and the r 'th coin toss position as successors. The game is very similar to a simple stochastic game used as an example by Condon [7] to show that unmodified value iteration converges slowly.

In this subsection we will show that $E_{n,r}$ is an extremal game in the sense of Definition 5 and upper bound $T(E_{n,r})$, thereby upper bounding $T(G)$ for all $G \in S_{n,r}$.

The first two lemmas in this subsection concerns assumptions about t -extremal games we can make without loss of generality.

Lemma 8 *For all n, r, t , there is a t -extremal game in $S_{n,r}$ with $V_1 = \emptyset$, i.e., without containing positions belonging to Max.*

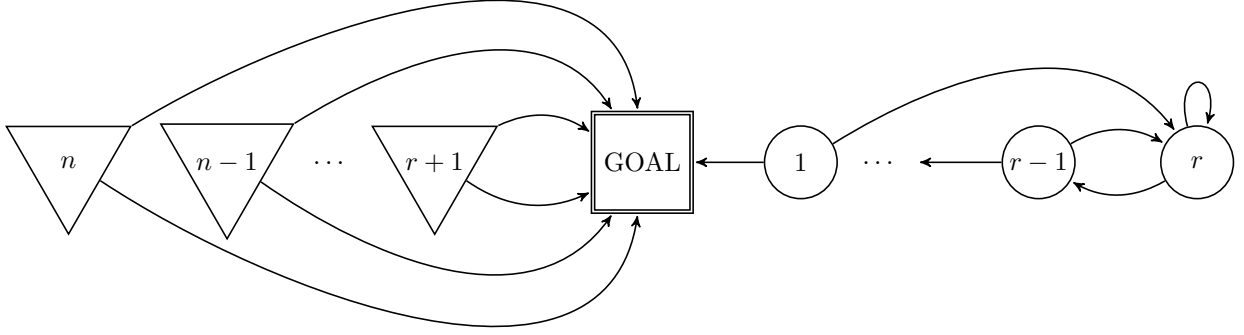


Figure 3: The extremal game $E_{n,r}$. Circle nodes are coin toss positions, triangle nodes are Min positions and the node labeled GOAL is the GOAL position.

Proof Take any t -extremal game $G \in S_{n,r}$. Let x be an optimal positional strategy for Max in this game. Now replace each position belonging to Max with a position belonging to Min with both outgoing arcs making the choice specified by x . Call the resulting game H . We claim that H is also t -extremal. First, clearly, each position k of H has the same value as is has in G , i.e., $\text{val}(G)_k = \text{val}(H)_k$. Also, if we compare the values of the positions of the games H^t and G^t defined in the statement of Lemma 3, we see that $\text{val}(H^t)_k \leq \text{val}(G^t)_k$, since the only difference between H^t and G^t is that player Max has more options in the latter game. Therefore, $\text{val}(H)_k - \text{val}(H^t)_k \geq \text{val}(G)_k - \text{val}(G^t)_k$ so H must also be t -extremal. \square

Lemma 9 For all n, r, t , there exists a t -extremal game in $S_{n,r}$, where all positions have value one and where no positions belong to player Max.

Proof By Lemma 8, we can pick a t -extremal game G in $S_{n,r}$ where no positions belong to player Max. Suppose that not all positions in G have value 1. Then, it is easy to see that the set of positions of value 0 is non-empty. Let this set be N . Let H be the game where all arcs into N are redirected to GOAL. Clearly, all positions in this game have value 1. We claim that H is also t -extremal.

Fix a position k . We shall show that $\text{val}(H)_k - \text{val}(H^t)_k \geq \text{val}(G)_k - \text{val}(G^t)_k$ and we shall be done. Let σ_k be a (not necessarily positional) optimal strategy for player Min in G^t for plays starting in k and let the probability measure on plays of G^t associated with this strategy be denoted Pr_{σ_k} . As σ_k is also a strategy that can be played in G , we have $\text{Pr}_{\sigma_k}[\text{Play does not reach } N] \geq \text{val}(G)_k$. Also, by definition, $\text{Pr}_{\sigma_k}[\text{Play reaches GOAL}] = \text{val}(G^t)_k$. That is,

$$\text{Pr}_{\sigma_k}[\text{Play reaches neither GOAL nor } N] \geq \text{val}(G)_k - \text{val}(G^t)_k.$$

Let $\bar{\sigma}_k$ be an optimal strategy for plays starting in k for player Min in H^t . This strategy can also be used in G^t . Let the probability distribution on plays of G^t associated with this strategy be denoted $\text{Pr}_{\bar{\sigma}_k}$. Note that plays reaching GOAL in H^t correspond to those plays reaching either GOAL or

N in G^t . Thus, by definition, $\Pr_{\bar{\sigma}_k}[\text{Play reaches neither GOAL nor } N] = 1 - \text{val}(H^t)_k$. As σ_k can be used in H^t where $\bar{\sigma}_k$ is optimal, we have

$$1 - \text{val}(H^t)_k \geq \text{val}(G)_k - \text{val}(G^t)_k.$$

But since $\text{val}(H)_k = 1$, this is the desired inequality

$$\text{val}(H)_k - \text{val}(H^t)_k \geq \text{val}(G)_k - \text{val}(G^t)_k.$$

□

The next lemma will be used to derive a ordering of the positions in any game G satisfying the restrictions of Lemma 9.

Lemma 10 *Let G be a game without Max positions in which all positions have value one. Let V' be a non-empty set of positions of G that does not include GOAL. Then, at least one of the following two cases hold:*

1. V' contains a Min position with both successors outside of V' or
2. V' contains a coin toss position with at least one successor outside of V' .

Proof Suppose not. Then the Min-player can force play to stay within V' when play starts in V' . Thus, the values of all positions in V' are 0, a contradiction. □

The following lemma will be used several times to change the structure of a game while only making it more extremal, eventually making the game into the specific game $E_{n,r}$ (in the context of extremal combinatorics, this is a standard technique pioneered by Moon and Moser [14]).

Lemma 11 *Given a game G . Let c be a coin toss position in G and let k be an immediate successor position k of c . Also, let a position k' with the following property be given: $\forall t : \text{val}(G^t)_{k'} \leq \text{val}(G^t)_k$. Let H be the game where the arc from c to k is redirected to k' . Then, $\forall t, j : \text{val}(H^t)_j \leq \text{val}(G^t)_j$.*

Proof In this proof we will throughout refer to the properties of **ModifiedValueIteration** and use Lemma 3. We show by induction in t that $\forall j, t : \text{val}(H^t)_j \leq \text{val}(G^t)_j$. For $t = 0$ we have $\text{val}(H^t)_j = \text{val}(G^t)_j$ by inspection of the algorithm. Now assume that the inequality holds for all values smaller than t and for all positions i and we will show that it holds for t and all positions j . Consider a fixed position j . There are three cases.

1. *The position j belongs to Max or Min.* In this case, we observe that the function computed by **SolveDGG** to determine the value of position j in **ModifiedValueIteration** is a monotonously increasing function. Also, the deterministic graphical game obtained when replacing coin toss positions with terminals is the same for G and for H . By the induction hypothesis, we have that for all i , $\text{val}(H^{t-1})_i \leq \text{val}(G^{t-1})_i$. So, $\text{val}(H^t)_j \leq \text{val}(G^t)_j$.
2. *The position j is a coin toss position, but not c .* In this case, we have

$$\text{val}(G^t)_j = \frac{1}{2}\text{val}(G^{t-1})_a + \frac{1}{2}\text{val}(G^{t-1})_b,$$

and

$$\text{val}(H^t)_j = \frac{1}{2}\text{val}(H^{t-1})_a + \frac{1}{2}\text{val}(H^{t-1})_b$$

where a and b are the successors of j . By the induction hypothesis, $\text{val}(H^{t-1})_a \leq \text{val}(G^{t-1})_a$ and $\text{val}(H^{t-1})_b \leq \text{val}(G^{t-1})_b$. Again, we have $\text{val}(H^t)_j \leq \text{val}(G^t)_j$.

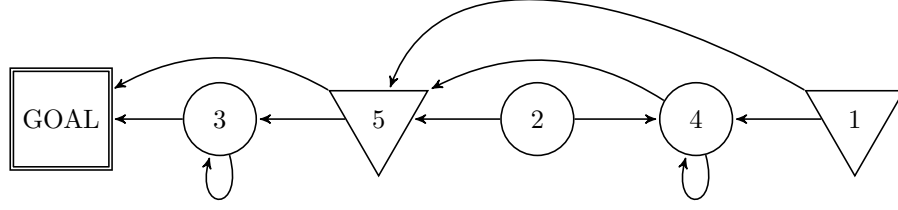


Figure 4: An example of H_0 , with $n = 5$ and $r = 3$. Circle nodes are coin toss positions, triangle nodes are Min positions and the node labeled GOAL is the GOAL position. Note that this particular H_0 is not extremal.

3. *The position j is equal to c .* In this case, we have

$$\text{val}(G^t)_c = \frac{1}{2}\text{val}(G^{t-1})_a + \frac{1}{2}\text{val}(G^{t-1})_k$$

where a and k are the successors of c in G while

$$\text{val}(H^t)_c = \frac{1}{2}\text{val}(H^{t-1})_a + \frac{1}{2}\text{val}(H^{t-1})_{k'}.$$

By the induction hypothesis we have $\text{val}(H^{t-1})_a \leq \text{val}(G^{t-1})_a$. We also have that $\text{val}(H^{t-1})_{k'} \leq \text{val}(G^{t-1})_{k'}$ which is, by assumption, at most $\text{val}(G^{t-1})_k$. So, we have $\text{val}(H^t)_c \leq \text{val}(G^t)_c$. \square

Theorem 12 $E_{n,r}$ is an extremal game in $S_{n,r}$.

Proof Let $H_0 \in S_{n,r}$, where $V_1 = \emptyset$ and all positions in H_0 have value one. We will show that for all t we have that $\max_k(\text{val}(E_{n,r})_k - \text{val}(E_{n,r}^t)_k) \geq \max_{k'}(\text{val}(H_0)_{k'} - \text{val}(H_0^t)_{k'})$. Since by Lemma 9, we can take H_0 to be a t -extremal game for any t , $E_{n,r}$ is a t -extremal game for all t and is hence an extremal game.

To illustrate the proof we will use as running example the game in Figure 4.

We shall construct a sequence H_1, H_2, \dots, H_n of games (which will in fact be identical to H_0 , except that positions have been renumbered), so that H_k has the following property P_k .

Property P_k : Any Min position j among the positions $1, 2, \dots, k$ has *all* successors within the set $\{1, 2, \dots, j-1, \text{GOAL}\}$. Any coin toss position j among the positions $\{1, 2, \dots, k\}$ has *at least one* successor within $\{1, 2, \dots, j-1, \text{GOAL}\}$.

Suppose we already constructed H_j for $j < k$. We show how to construct H_k based on H_{k-1} . Applying Lemma 10 to the game H_{k-1} with $V' = \{k, \dots, n\}$, we find among the positions k, \dots, n either a coin toss position u with one successor in $\{1, 2, \dots, k-1, \text{GOAL}\}$ or a Min-position u with all successors in $\{1, 2, \dots, k-1, \text{GOAL}\}$. In either case, we renumber u to k and k to u and let the resulting game be H_k .

Figure 5 shows H_n for the case of our running example.

Each coin toss position in H_n has at least one successor with a lower index than itself. (Recall that GOAL has index 0.) In the following, we call this successor the *lower* successor and the other successor the *higher* successor. If both successors in fact have a lower index than the position, we choose one of the two successor arbitrarily as the higher successor.

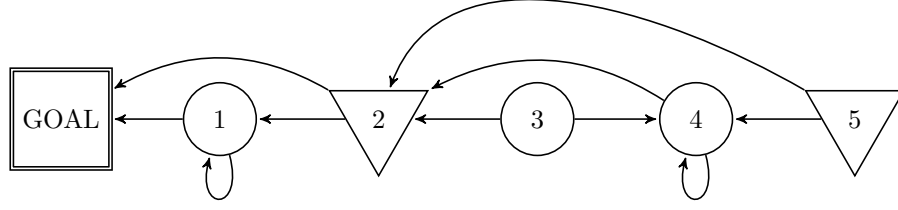


Figure 5: H_n , if H_0 is the game in Figure 4.

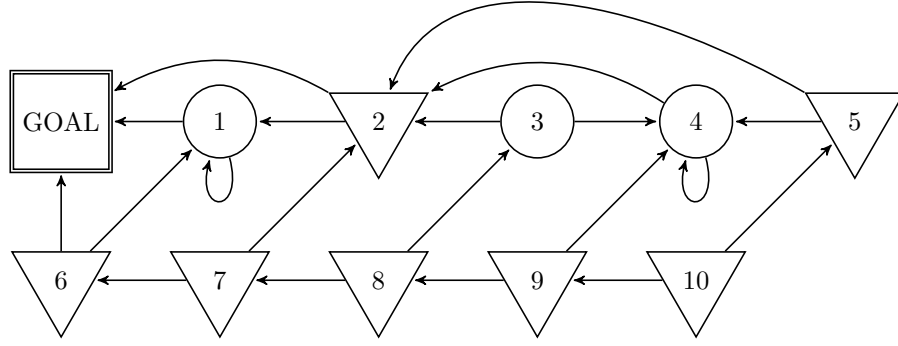


Figure 6: H' , if H_0 is the game in Figure 4.

We now make a series of transformations from H_n generating a new sequence of games. This will take us outside the set of games in $S_{n,r}$, but the last game in the sequence will again be in $S_{n,r}$. For each transformation, from G' to G'' , we will show that $\text{val}(G''^t)_k \leq \text{val}(G'^t)_k$ for all t and k . For the final game $E_{n,r}$ we arrive at, we clearly have that $\text{val}(E_{n,r})_k = 1$ for all k . This is in fact also true for all intermediate games, but we shall not need that fact.

For each of the original non-terminal positions $1, 2, \dots, n$ in H_n , we add a Min-position. We assign index $n + j$ to the Min-position associated with position j . We let the two successors of position $n + j$ be j and $n + j - 1$, except for the case of $n + 1$, where we let the two successors be 1 and GOAL. Let the resulting game be denoted H' . For our running example, H' is shown in Figure 6.

Applying Lemma 3 and inspecting the code of `ModifiedValueIteration`, we have that

$$\forall k \in \{1, 2, \dots, n\}, t : \text{val}(H'^t)_k = \text{val}(H_n^t)_k.$$

We will only use that $\forall k \in \{1, 2, \dots, n\}, t : \text{val}(H'^t)_k \leq \text{val}(H_n^t)_k$.

We also have the following fact: $\text{val}(H'^t)_{2n} = \min_{j \in \{1, 2, \dots, n\}} \text{val}(H'^t)_j$. Indeed, we can argue by induction in k that $\text{val}(H'^t)_{n+k} = \min_{j \in \{1, 2, \dots, k\}} \text{val}(H'^t)_j$. For the base case, we have that $\text{val}(H'^t)_{n+1} = \min(1, \text{val}(H'^t)_1)$. For $j > 1$, we have

$$\text{val}(H'^t)_{n+j} = \min(\text{val}(H'^t)_j, \text{val}(H'^t)_{n+j-1}),$$

completing the proof. This property, and the fact that the proof only used information about the successors of $n + k$ for $k \in \{1, 2, \dots, n\}$, allows us to apply Lemma 11 iteratively to modify the

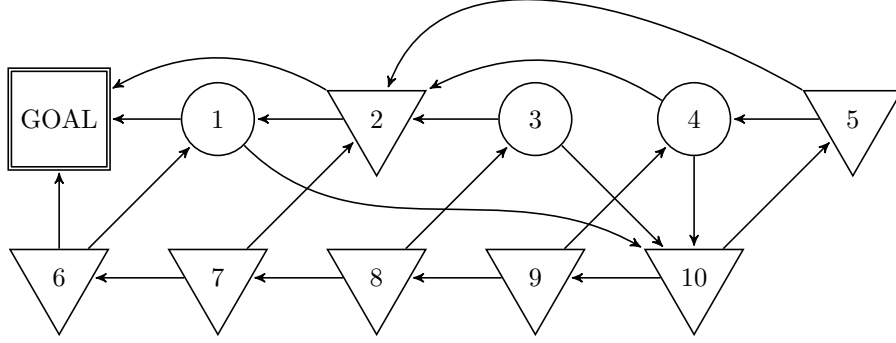


Figure 7: H'_0 , if H_0 is the game of Figure 4.

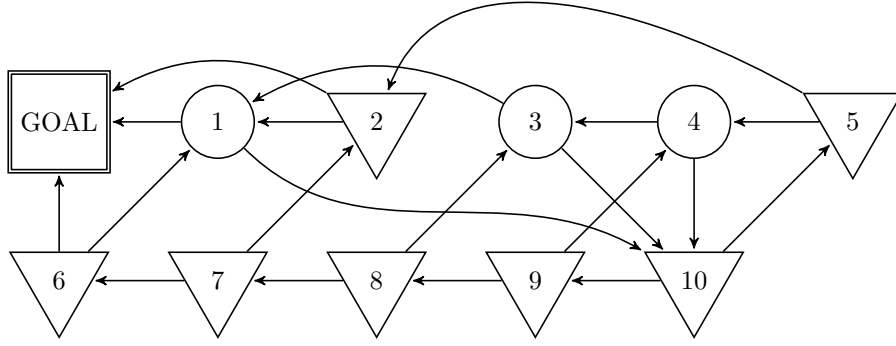


Figure 8: H'_r , if H_0 is the game in Figure 4.

game by changing the higher successor of each and every coin toss position to be the position $2n$. We denote this game H'_0 .

For our running example, H'_0 is shown in Figure 7.

Let the coin toss positions in H'_0 be $i_1 < i_2 < \dots < i_r$. Then, we define a sequence of games H'_1, H'_2, \dots, H'_r as follows. We define H'_1 from H'_0 by changing the lower successor of i_1 to GOAL. For $j > 1$, we define H'_j from H'_{j-1} by changing the lower successor of i_j to be i_{j-1} .

For our running example, H'_r is given in Figure 8.

Claim. For $t \geq 0, j \in \{1, \dots, r+1\}$, the following holds. For a position with index k strictly smaller than i_j , we have $\text{val}(H'^t_{j-1})_k \geq \text{val}(H'^t_{j-1})_{i_{j-1}}$. Here, by convention, we let i_0 be the GOAL position when considering the statement for $j = 1$ and we let i_{r+1} be ∞ when considering the statement for $j = r+1$.

Proof of claim. The proof is by induction in j .

Clearly, $\text{val}(H'^t_{j'})_k = 1$ for all positions k in $1, 2, \dots, i_1 - 1$ and for all j' and t , so this settles the base case of $j = 1$.

For larger values of j , and $k < i_{j-1}$, we have by construction that $\text{val}(H'^t_{j-1})_k = \text{val}(H'^t_{j-2})_k$. Now, $k \geq i_{j-1}$. Therefore there are two cases. Either k is a coin toss position or k is a Min-position.

If k is a coin toss position, we can without loss of generality assume that $k = i_{j-2}$, since

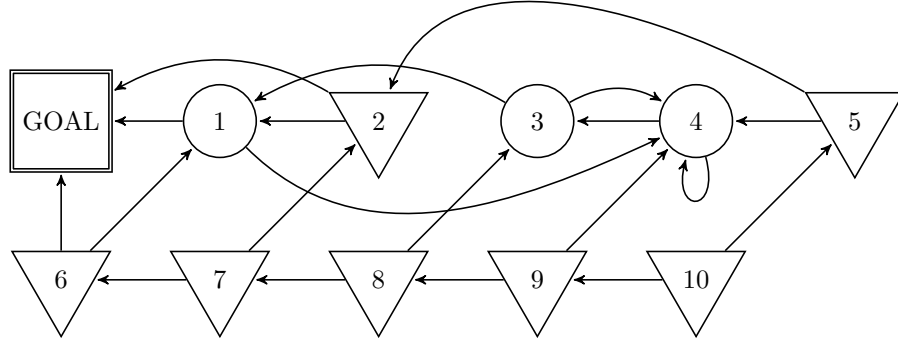


Figure 9: H'' , if H_0 is the game in Figure 4.

it has the smallest value among all coin toss positions, by the induction hypothesis. Therefore, we only need to show that $\forall t : \text{val}(H_{j-1}^t)_{i_{j-2}} \geq \text{val}(H_{j-1}^t)_{i_{j-1}}$. For $j = 2$, we have that $\forall t : \text{val}(H_1^t)_{i_0} = 1 \geq \text{val}(H_1^t)_{i_1}$. For $j \geq 3$, we have by the properties of the algorithm that for $\text{val}(H_{j-1}^0)_{i_{j-2}} = 0 = \text{val}(H_{j-1}^0)_{i_{j-1}}$ and $\forall t \geq 1 : \text{val}(H_{j-1}^t)_{i_{j-2}} = \frac{1}{2}\text{val}(H_{j-1}^{t-1})_{i_{j-3}} + \frac{1}{2}\text{val}(H_{j-1}^{t-1})_{i_{2n}}$. We have by the induction hypothesis that

$$\begin{aligned} \frac{1}{2}\text{val}(H_{j-1}^{t-1})_{i_{j-3}} + \frac{1}{2}\text{val}(H_{j-1}^{t-1})_{i_{2n}} &\geq \frac{1}{2}\text{val}(H_{j-1}^{t-1})_{i_{j-2}} + \frac{1}{2}\text{val}(H_{j-1}^{t-1})_{i_{2n}} \\ &= \text{val}(H_{j-1}^t)_{i_{j-1}}. \end{aligned}$$

If k is a Min-position in $\{i_{j-1} + 1, \dots, i_j - 1\}$, assume to the contrary that some k fails to satisfy $\text{val}(H_{j-1}^t)_k \geq \text{val}(H_{j-1}^t)_{i_{j-1}}$. Consider the smallest such k . As k is a Min-position with two successors both of which are smaller than k , we have that $\text{val}(H_{j-1}^t)_k$ is the minimum of two numbers, both of which are at least $\text{val}(H_{j-1}^t)_{i_{j-1}}$, either by the induction hypothesis or the assumption that k is minimal. *This completes the proof of the claim.*

By the claim, for all positions $k \in \{1, 2, \dots, n\}$, we have that $\forall t \geq 0 : \text{val}(H_r^t)_k \geq \text{val}(H_r^t)_{i_r}$. Also, by an induction argument identical to the one we used to argue a similar property for H' , we have $\forall t \geq 0 : \text{val}(H_r^t)_{2n} = \text{val}(H_r^t)_{i_r}$. Thus we may define the game H'' from H'_r by applying Lemma 11 and changing all higher successors of all coin toss positions i_j to be i_r (instead of $2n$).

For our running example, H'' is the game of Figure 9.

Finally, we arrive at $E_{n,r}$ by subsequently removing all “new” Min-positions $n + 1, \dots, 2n$ and changing all successors of all original Min-positions of H'' to be the GOAL position.

For our running example, $E_{5,3}$ is the extremal game identified by the proof, and is given in Figure 10. Note that the same game would also be identified for any other game with $r = 3$ and $n = 5$ (up to the indexing of the positions).

It is easy to see that all positions in $E_{n,r}$ have value 1. We have that each Min-position k in H'' satisfies $\text{val}(H''^t)_k \geq \text{val}(H''^t)_{i_r}$.

We have that $\text{val}(H_n^t)_k \geq \text{val}(H''^t)_{i_r} = \text{val}(E_{n,r}^t)_{i_r}$ for all $k \in \{1, 2, \dots, n\}$, since we found H'' either by applying Lemma 11 or in a way that did not change the value of any position for any time bound.

Also, $E_{n,r} \in S_{n,r}$, therefore, since at least one possible option for $H_0 \in S_{n,r}$ (and therefore H_n ,

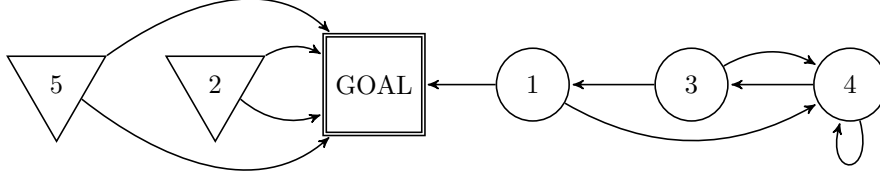


Figure 10: The game $E_{5,3}$, upto indexing of the positions.

since H_n was a reindexing of the positions in H_0) was a t -extremal game for any t , $E_{n,r}$ is extremal. This completes the proof of the lemma. \square

Having identified the extremal game $E_{n,r}$, we next estimate $T(E_{n,r})$.

Lemma 13 For $\forall n, r > 0, \epsilon > 0, t \geq 2(\ln 2\epsilon^{-1}) \cdot 2^r, k \in V : \text{val}(E_{n,r})_k - \text{val}(E_{n,r}^T)_k \leq \epsilon$

Proof We observe that for the purposes of estimating $\text{val}(E_{n,r})_k - \text{val}(E_{n,r}^T)_k$, we can view $E_{n,r}$ as a game containing r coin toss positions only, since all Min-positions point directly to GOAL. Also, when modified value iteration is applied to a game G containing only coin toss positions, Lemma 3 implies that $\text{val}(G^t)_k$ can be reinterpreted as the probability that the absorbing Markov process starting in state k is absorbed within t steps. By the structure of $E_{n,r}$, this is equal to the probability that a sequence of t fair coin tosses contains r consecutive tails. This is known to be exactly $1 - F_{t+2}^{(r)}/2^t$, where $F_{t+2}^{(r)}$ is the $(t+2)$ 'nd Fibonacci r -step number, i.e. the number given by the linear homogeneous recurrence $F_m^{(r)} = \sum_{i=1}^r F_{m-i}^{(r)}$ and the boundary conditions $F_m^{(k)} = 0$, for $m \leq 0$, $F_1^{(r)} = F_2^{(r)} = 1$. Asymptotically solving this linear recurrence, we have that $F_m^{(r)} \leq (\phi_r)^{m-1}$ where ϕ_r is the root near 2 to the equation $x + x^{-r} = 2$. Clearly, $\phi_r < 2 - 2^{-r}$, so

$$F_{t+2}^{(r)}/2^t < \frac{(2 - 2^{-r})^{t+1}}{2^t} = 2(1 - 2^{-r-1})^{t+1} < 2(1 - 2^{-r-1})^t.$$

Therefore, the probability that the chain is not absorbed within $t = 2(\ln 2\epsilon^{-1}) \cdot 2^r$ steps is at most

$$2(1 - 2^{-r-1})^{2(\ln 2\epsilon^{-1}) \cdot 2^r} \leq 2e^{-\ln 2\epsilon^{-1}} = \epsilon.$$

\square

Corollary 14 For $\forall n, r > 0 : T(E_{n,r}) \leq 2(\ln 2^{5r+1}) \cdot 2^r$.

Proof The proof is by insertion into Lemma 13. \square

4 Conclusions

We have shown an algorithm solving simple stochastic games obtaining an improved running time in the worst case compared to previous algorithms, as a function of its number of coin toss positions.

It is relevant to observe that the *best* case complexity of the algorithm is strongly related to its *worst* case complexity, as the number of iterations of the main loop is fixed in advance.

As mentioned in the introduction, our paper is partly motivated by a result of Chatterjee *et al.* [4] analysing the *strategy iteration* algorithm of the same authors [5] for the case of simple stochastic games. We can in fact improve their analysis, using the techniques of this paper. We combine three facts:

1. ([5, Lemma 8]) For a game $G \in S_{n,r}$, after t iterations of the strategy iteration algorithm [5] applied to G , the (positional) strategy computed for Max guarantees a probability of winning of at least $\text{val}(\bar{G}^t)_k$ against any strategy of the opponent when play starts in position k , where \bar{G}^t is the game defined from G in Definition 4 of this paper.
2. For a game $G \in S_{n,r}$ and all k, t , $\text{val}(\bar{G}^{t(n-r+1)})_k \geq \text{val}(G^t)_k$. This is a direct consequence of the definitions of the two games, and the fact that in an optimal play, either a coin toss position is encountered at least after every $n - r + 1$ moves of the pebble, or never again.
3. Corollary 14 of the present paper.

These three facts together implies that the strategy iteration algorithm after $2(\ln 2^{5r+1}) \cdot 2^r(n-r+1)$ iterations has computed a strategy that guarantees the values of the game within an additive error of 2^{-5r} , for $r \geq 6$. As observed by Chatterjee *et al.* [4], such a strategy is in fact optimal. Hence, we conclude that their strategy iteration algorithm terminates in time $2^r n^{O(1)}$. This improves their analysis of the algorithm significantly, but still yields a bound on its worst case running time inferior to the worst case running time of the algorithm presented here. On the other hand, unlike the algorithm presented in this paper, their algorithm has the desirable property that it may terminate faster than its worst case analysis suggests.

References

- [1] D. Andersson, K. A. Hansen, P. B. Miltersen, and T. B. Sørensen. Deterministic graphical games revisited. In *CiE '08: Proceedings of the 4th conference on Computability in Europe*, pages 1–10, Berlin, Heidelberg, 2008. Springer-Verlag.
- [2] D. Andersson and P. B. Miltersen. The complexity of solving stochastic games on graphs. In *Algorithms and Computation, 20th International Symposium, ISAAC 2009, Honolulu, Hawaii, USA, December 16-18, 2009. Proceedings*, volume 5878 of *Lecture Notes in Computer Science*, pages 112–121. Springer, 2009.
- [3] R. E. Bellman. On the application of dynamic programming to the determination of optimal play in chess and checkers. *Proceedings of the National Academy of Sciences of the United States of America*, 53:244–246, 1965.
- [4] K. Chatterjee, L. de Alfaro, and T. A. Henzinger. Termination criteria for solving concurrent safety and reachability games. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'09*, pages 197–206, 2009.
- [5] Krishnendu Chatterjee, Luca de Alfaro, and Thomas A. Henzinger. Strategy improvement for concurrent reachability games. In *Third International Conference on the Quantitative Evaluation of Systems. QEST'06.*, pages 291–300. IEEE Computer Society, 2006.
- [6] A. Condon. The complexity of stochastic games. *Information and Computation*, 96:203–224, 1992.

- [7] A. Condon. On algorithms for simple stochastic games. In *Advances in Computational Complexity Theory, volume 13 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 51–73. American Mathematical Society, 1993.
- [8] D. Dai and R. Ge. New results on simple stochastic games. In *Algorithms and Computation, 20th International Symposium, ISAAC 2009, Honolulu, Hawaii, USA, December 16-18, 2009. Proceedings*, volume 5878 of *Lecture Notes in Computer Science*, pages 1014–1023. Springer, 2009.
- [9] D. Gillette. Stochastic games with zero stop probabilities. In M. Dresner, A.W. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games III*, volume 39 of *Annals of Mathematics Studies*, pages 179–187. Princeton University Press, 1957.
- [10] H. Gimbert and F. Horn. Simple Stochastic Games with Few Random Vertices are Easy to Solve. In *Proceedings of the 11th International Conference on the Foundations of Software Science and Computational Structures, FoSSaCS’08*, volume 4962 of *Lecture Notes in Computer Science*, pages 5–19. Springer-Verlag, 2008.
- [11] S. Kwek and K. Mehlhorn. Optimal search for rationals. *Inf. Process. Lett.*, 86(1):23–26, 2003.
- [12] T. M. Liggett and S. A. Lippman. Stochastic games with perfect information and time average payoff. *SIAM Review*, 11(4):604–607, 1969.
- [13] J. F. Mertens and A. Neyman. Stochastic games. *International Journal of Game Theory*, 10:53–66, 1981.
- [14] J. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3:23–28, 1965. 10.1007/BF02760024.
- [15] L. S. Shapley. Stochastic games. *Proc. Nat. Acad. Science*, 39:1095–1100, 1953.